UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/606,089	06/25/2003	Brian S. Christian	MS1-1512US	4285

22971 7590 04/08/2008

MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052-6399

EXAMINER	
WILLIAMS, JEFFERY L	
ART UNIT	PAPER NUMBER
2137	

NOTIFICATION DATE	DELIVERY MODE
04/08/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

roks@microsoft.com
ntovar@microsoft.com
a-rydore@microsoft.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/606,089
Filing Date: June 25, 2003
Appellant(s): CHRISTIAN ET AL.

J. Richard Bucher
Reg. No. 57,971
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/14/08 appealing from the Office action mailed 5/29/07.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Scott, D. and Sharp, R. "Abstracting application-level web security". In Proceedings of the 11th international Conference on World Wide Web (Honolulu, Hawaii, USA, May 07 - 11, 2002). WWW '02. ACM, New York, NY, pg. 1 - 12.

For the reasons below, it is believed that the rejections should be sustained.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1, 4-12, 16-21, and 24-28 are rejected under 35 U.S.C. 102(b) as being anticipated by Scott et al. (Scott), "Abstracting Application-Level Web Security".

Regarding claim 1, Scott discloses:

receiving data input through a web page from a client device (fig. 1, page 2, col. 1, par. 3-6)*; referencing a declarative module to determine a client input security screen to apply to the data input from the client device* (page 3, col. 2, par. 2)*;*

wherein the declarative module comprises:

a global section that includes at least one client input security screen that applies to any type of client input value (fig. 2; page 6, col. 1, par. 1, 2, par. 2, lines 9-13). Scott discloses input security screens (i.e. a transformation screen) that are applied to all user input (parameters values);

an individual values section that includes at least one client input security screen that applies to a particular type of client input value (fig. 2; page 4, col. 1). Herein, Scott discloses screens for screening particular types of client input values (i.e. cookies, urls, other parameters). Thus Scott discloses an individual values section.

and applying multiple client input security screens to the data input from the client device (page 3, col. 2, par. 2; fig. 2)*, including at least one client input security screen from the global section of the declarative module and at least one client input security screen from the individual values section of the declarative module, wherein the client input security screens are distinct from one another* (page 3, col. 2, par. 1, 2; fig. 2). Herein, Scott discloses separate screens.

and wherein said act of referencing comprises first using the global section to screen one or more client input values and then using the individual values section to screen at least one of said one or more client input values (sect. 3.4, par. 3).

Regarding claim 4, Scott discloses:

wherein the particular type of client input value is one of the following types of client input values: query string; server variable; form value; cookie (Scott, fig. 2).

Regarding claim 5, Scott discloses:

wherein the declarative module further comprises a web.config file (Scott, page 1, col. 2, par.3; page 3, col. 2, par. 1).

Regarding claim 6, Scott discloses:

wherein the applying the client input security screen further comprises executing a default action on invalid client input detected by the client input security screen (Scott, page 3, col. 2, par. 1, lines 8-13, par. 2, lines 5-11; page 4, col. 2, par. 3,4). Scott discloses the application of several types of input screening to all input data (default screening) wherein actions are performed on the all the input data during the process of data input security screening. Additionally, Scott discloses default transformations that can be applied during the screening of invalid input data.

Regarding claim 7, Scott discloses:

wherein the applying the client input security screen further comprises executing a specified action on invalid client input detected by the client input security screen, the specified action being specified in the client input security screen (Scott, page 4, col. 1, par. 4-6).

Regarding claim 8, Scott discloses:

wherein a client input security screen further comprises one or more values that may be entered as client input, the one or more values further comprising the only values that may be entered as client input (Scott, page 4, col. 1, par. 4-6). Scott discloses a security screen that constrains client input to a set of values, such as any integer: 0 – int [length 4]. Thus, the security screen effectively comprises the values of 0 – int [length 4] to be imposed upon the client input as a restriction. Additionally, Scott discloses that the security screen comprises specific URL values (extracted from HTTP requests) that may be entered as client input (Scott, page 6, col. 2, par. 1).

Regarding claim 9, Scott discloses:

wherein a client input security screen further comprises one or more screened values that, when detected in the client input, cause an action to be taken on the client input (Scott, fig. 4; page 3, col. 2, par. 2; page 4, col. 2, par. 3).

Regarding claim 10, Scott discloses:

wherein the action to be taken further comprises removing the one or more screened values detected in the client input (Scott, fig. 4; page 3, col. 2, par. 2; page 4, col. 2, par. 3, 4). Scott discloses the encoding of screened values (removal and replacement). Additionally, Scott discloses the removal of values from client input based upon the client input security screen (Scott, page 7, col. 2, par. 1.1 – 1.2)

Regarding claim 11, Scott discloses:

wherein the action to be taken further comprises removing an entire string that contains the one or more screened values detected in the client input (Scott, page 6, col. 2, par. 3; fig. 5; page 9, col. 1, par. 2.2).

Regarding claim 12, it is the system claim corresponding to the method claim 1, and is rejected for, at least, the same reasons, and furthermore because Scott discloses:

a web page server unit configured to provide one or more web pages to one or more client devices over a distributed network (Scott, fig. 1).

Regarding claim 16, Scott discloses:

wherein a screening rule further comprises a client input variable that may be accepted as input from a client (Scott, fig. 5). Scott discloses various screening rules that accept client input variables.

Regarding claim 17, Scott discloses:

wherein a screening rule further comprises one or more screened characters that, when detected in client input, are screened from the client input according to a screening rule (Scott, fig. 5 – see transformation).

Regarding claim 18, Scott discloses:

wherein the screening rule further comprises a default screening action that is applied in the absence of a specified screening action (Scott, fig. 5 – see transformation). Scott discloses a single screening action that is to be performed, and thus, a default screening action.

Regarding claim 19, Scott discloses:

wherein the screening rule further comprises a specified screening action that is applied to the screened client input (Scott, fig. 5 – see transformation). Scott discloses a single specific screening action that is to be performed.

Regarding claim 20, it is rejected, at least, for the same reasons as claim 5.

Regarding claim 21, it is rejected, at least, for the same reasons as claim 1, and furthermore because Scott discloses:

serving a web page to a client over a distributed network; receiving client input via the web page (Scott, fig. 1, page 2, col. 1, par. 3-6)*; comparing the client input with multiple and distinct client input security screens stored in a security declarative module; wherein the security declarative module includes a global section configured to screen all types of client input values and an individual values section configured to screen particular types of client input values* (see rejection of claim 1)*; if invalid client input is detected, performing a screening action on the invalid client input as indicated by the*

security declarative module (Scott, page 3, col. 2, par. 2; page 4, col. 2, par. 3; page 6, col. 1, par. 1, 2; fig. 5)*; and wherein the client input security screens included in the security declarative module can be applied to multiple web pages* (Scott, page 4, col. 1, par. 2).

Furthermore, Scott discloses a computer system, and thus discloses media and instructions (Scott, fig. 1).

Regarding claims 24 and 25, they are the media and instruction claims corresponding to the method and system claims of 5 – 7, 18, and 19, and they are rejected for, at least, the same reasons.

Regarding claim *26*, Scott discloses:

wherein the screening action further comprises a default action that is not required to be specified in a client input security screen (Scott, page 6, col. 1, par. 1, 2).

Regarding claims *27* and 28, Scott discloses:

wherein the multiple web pages are included in a web project and *wherein the multiple web pages are included in a web-based application* (Scott, Abstract; Introduction; fig. 1; section 3.1; page 4, col. 1, par. 2; page 6, col. 1, par. 2, col. 2, par. 1). Scott discloses a security policy to be applied to a large web-application, the policy comprising rules for the web pages of a site. The web pages are associated with a web application, thus, they are included in a web project/application.

(10) Response to Argument

Appellant's arguments filed 1/14/08 have been fully considered but they are not persuasive.

(i) Appellant contends that Scott fails to disclose "referencing a declarative security module" (Appeal Brief, pg. 12, par. 2)

First, Page 3 (Col. 2 - Para. 2) of Scott merely describes a policy compiler responsible for generating SPDL code which is loaded into a security gateway which acts as a firewall. Missing is any discussion of "referencing a declarative module..." as claimed. As such, Applicant submits that the Office's reliance on this excerpt is misplaced.

In response, the examiner respectfully notes that page 3, col. 2, par. 2 of Scott discloses more than "*merely … a policy compiler responsible for generating SPDL code which is loaded into a security gateway which acts as a firewall*". Scott discloses the determination of validation constraints and transformation rules ["client input security screens"] that are to be applied to input received from the client. This determination is made by "referencing" [see also Scott, sect. 3.3, par. 1, lines 1-3] a SPDL policy [i.e. an XML specification and corresponding DTD (the specification and DTD containing statements which declare the security screening techniques to be performed) – thus, a

"declarative module"] – also see Scott, pg. 4, col. 1, par. 2; fig. 5). Therefore, Scott discloses "referencing a declarative security module".

(ii) Appellant contends that Scott fails to disclose "a global section". (Appeal Brief, pg. 12, par. 3)

Second, Fig. 2 and Page 6 (Col. 1 - Paras. 1-2) of Scott simply fail to disclose "a global section..." as claimed.

In response, the examiner respectfully notes that Scott discloses that the security policy comprises a section of statements for screening all client input, and thus teaches "a global section". See Scott, page 6, col. 1, par. 1, 2, par. 2, lines 9-13; fig. 2. Herein, Scott discloses input security screens (i.e. a transformation screen) that are applied to all user input (parameters values). See also, Scott, pg. 2, col. 2, "Cross-Site-Scripting" section, par. 3.

(iii) Appellant contends that Scott fails to disclose "first using the global section". (Appeal Brief, pg. 13, par. 2)

Finally, even if Scott did disclose "a global section..." as claimed, which it does not, Section 3.4 fails to disclose "...first using the global section" as claimed - and instead actually teaches away from this subject matter.

In response, the examiner respectfully notes that Scott clearly discloses first using the "global section" (e.g. the encoding transformation of all input) then using "the individual values section" (i.e. validating individual parameters). For example, figure 4 distinctly shows the operational process, wherein the sequence is "Apply transformations"(FIRST) → "Execute Validation Code" (SECOND). Also, paragraph 3 of section 3.4 reveals the sequence of using the declared transformations and validations. Within the sequence shown in paragraph 3, the transformations occur before the validations. If one event comes before another event within a sequence, the event that comes before the other is said to be first within that sequence.

(iv) Appellant contends that Scott fails to disclose a "web.config" file. (Appeal Brief, pg. 14, par. 4)

Additionally, regarding claim 5, which recites "...wherein the declarative module further comprises a web.config file", the excerpts cited by the Office on Pages 1 and 3 of Scott describe a special security policy description language (SPDL) used to write security policies. These excerpts do not disclose or suggest "wherein the declarative module further comprises a web.config file."

In response, the examiner respectfully notes that Scott discloses providing a security policy for a web application (such as a ASP or PHP applications - see Scott, sect. 3.5.2; fig. 5). The security policy comprises an XML file containing statements

declaring the various transformation and validation security screens for configuring the security of a web application. Thus, Scott discloses a "web.config" file for a web application, which according to the appellants is a file comprising the security screening statements for a web application and is "designated "web.config."" (See Appellants' specification, pg. 5, lines 15-23; pg. 6, line 23 - pg. 7, line 1). The examiner respectfully points out that, whether or not the appellants designate a file as "web.config" or any other name or title, there remains no structural or functional difference between the claimed file containing security screening statements and the prior art file comprising the security screening statements.

(v) Appellant contends that Scott fails to disclose "executing a default action" and "executing a specified action." (Appeal Brief, pg. 15, par. 1)

These excerpts do not disclose or suggest "executing a default action" or "executing a specified action" as claimed.

In response, the examiner respectfully asserts there is no difference between recitations of "default" or "specified" actions and the actions performed by the system of Scott.

Specifically, the examiner points out that any action performed by a *designed* system, such as a computing system, is a "specified" action. In other words, a designed system depends upon the specification of the designer for operation. Thus, the actions performed by such a system (e.g. '*upon event X perform action Y*' or '*upon event X*

generate error' or *'upon event X do something random'* or *'upon event X be passive/be active'*) are all "specified" ways for a computer to act as determined by the designer.

Throughout the disclosure of Scott, it may be seen that the system of Scott comprises the application of a plurality of client input security screens upon valid and invalid input. For example, see the screening performed upon client input as shown in Fig. 4 ("Check URL and parameter names", "Type Checking", "Check MAC", "Apply transformations", and "Execute Validation Code"). The system of Scott performs various actions, as specified by the system designer, upon such input data (invalid or valid) such as placing constraints upon the input data, preventing the input data's propagation to the server and returning error pages, encoding the input data, performing calculations upon the input data (i.e. MAC), and filtering the responses corresponding to the input data (Scott, pg. 3, col. 2, par. 2; pg. 4, col. 1, par. 2, 5; pg. 4, col. 2, par. 2, 3; pg. 6, col. 2, par. 1, 2, 3; pg. 7, col. 1, par. 5; pg. 8, col. 2, par. 1).

Furthermore, regarding "default" actions, the examiner respectfully notes that Scott discloses an automated system (Scott, fig. 1; fig. 4). The system of Scott operates according to a defined design. Thus, the actions performed by such a system constitute a default operation of the system. While it is noted that automated systems possibly may comprise means for a user to override the default operation or actions of the system, the examiner respectfully points out that Scott does not disclose such a manual overriding of the system operation (see for example, Scott, fig. 4). Thus, the system of Scott clearly discloses "default" actions.

(vi) Appellant contends that Scott fails to disclose "removing an entire string that contains the one or more screened values detected in the client input." (Appeal Brief, pg. 15, par. 3)

Additionally, regarding claim 11, which recites [t]he method as recited in claim 9, wherein the action to be taken further comprises removing an entire string that contains the one or more screened values detected in the client input", the excerpts/figure cited by the Office on Pages 6 and 9, and in Fig. 5, of Scott merely describe/depict transformation(s) and fail to disclose or suggest "...removing an entire string that contains the one or more screened values detected in the client input" as claimed.

In response, the examiner respectfully notes that Scott discloses the encoding of all input data. Thus, Scott discloses removing all the original data and replacing the original data with substituted characters or numbers (see for example, Scott, page 6, col. 1, lines 1 - 12). As such, Scott discloses *"...removing an entire string that contains the one or more screened values detected in the client input".*

(vii) Appellant contends that Scott fails to disclose the features of claims 12, 16-20, 21, and 24-28. (Appeal Brief, pg. 16 - 22)

In response, the examiner respectfully notes that claims 12, 16-20, 21, and 24-28 are system and computer program claims corresponding to the method claims of 1 and 4 – 11. As the appellant repeats what are essentially the same arguments as for

method claims 1 and 4 – 11, the examiner respectfully notes that such arguments regarding the system and computer program claims appear to be unpersuasive for the same reasons as shown above.

(viii) Appellant contends that Scott fails to disclose "a default action that is not required to be specified in a client input security screen". (Appeal Brief, pg. 22, par. 1)

Additionally, regarding claim 26, which recites "...wherein the screening action further comprises a default action that is not required to be specified in a client input security screen"...

In response, the examiner respectfully notes that Scott discloses an automated system that performs actions (e.g. "return error page") which are not required to be specified within a client security screen (compare, for example, to fig. 5 – wherein the example declarative module does not comprise specifying a "return error page" as is performed when certain screenings fail). (Scott, pg. 6, col. 1, par. 1, 2, col. 2; fig. 4). Thus, Scott discloses disclose "a default action that is not required to be specified in a client input security screen".

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Respectfully submitted,

Jeffery Williams
/Jeffery Williams/
Examiner, Art Unit 2137

/Gilberto Barron Jr/
Supervisory Patent Examiner, Art Unit 2132

Conferees:

/G. B./
Supervisory Patent Examiner, Art Unit 2132

Benjamin Lanier
/Benjamin E Lanier/
Primary Examiner, Art Unit 2132